NetworkVisualization(igraph)

Network visualization using igraph library

Read Data

```
rm(list = ls())
graphics.off()
cat("\014")
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
# as.is default behaviour is to convert character variables to factors
links <- read.csv('Dataset1-Media-Example-EDGES.csv', as.is = TRUE)
nodes <- read.csv('Dataset1-Media-Example-NODES.csv', as.is = TRUE)

nrow(links); nrow(unique(links[,c("from", "to")]))
```

```
## [1] 52
```

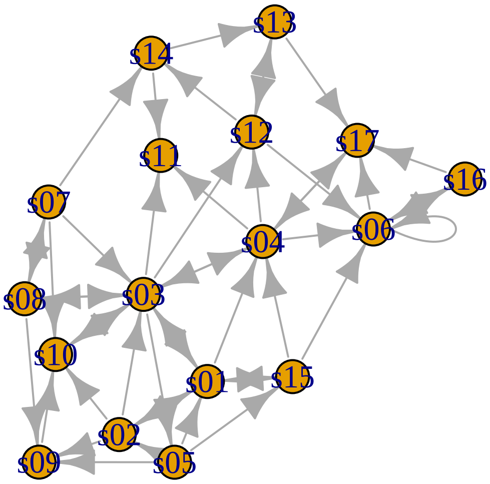
```
## [1] 49
```

```
links <- aggregate(links[,3], links[,-3], sum)
links <- links[order(links$from, links$to),]
colnames(links)[4] <- "weight"
rownames(links) <- NULL

nodes2 <- read.csv("Dataset2-Media-User-Example-NODES.csv", header=T, as.is=T)
links2 <- read.csv("Dataset2-Media-User-Example-EDGES.csv", header=T, row.names=1)
links2 <- as.matrix(links2)

# D:
# describes the edges of the network.
# Its first two columns are the IDs of the source and the target node for each edge.
# The following columns are edge attributes (weight, type, label, or anything else)
# VERTICES:
# starts with a column of node IDs.
# Any following columns are interpreted as node attributes.
net <- graph_from_data_frame(d=links, vertices=nodes, directed=T)
```

Developing Graph

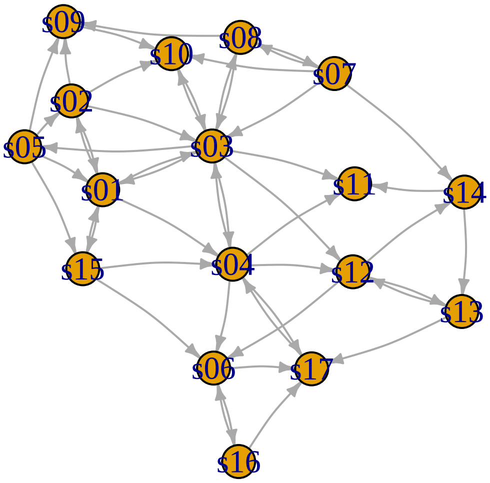
1st plot:

```
plot(net)
```



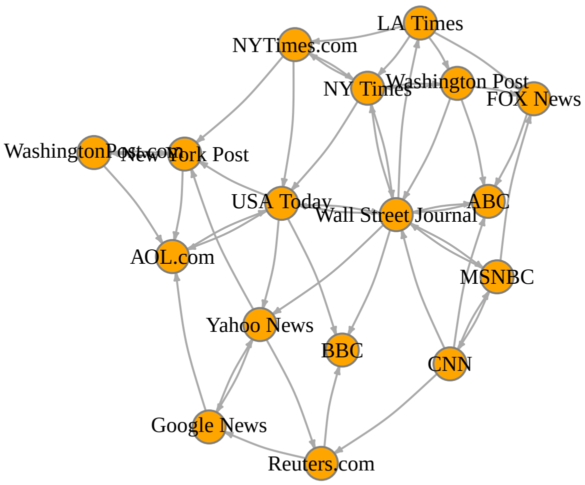
### 2nd plot:

```
##### removing loops in the graph #####
net <- simplify(net, remove.multiple = F, remove.loops = T)
# Plot with curved edges (edge.curved=.1) and reduce arrow size:
plot(net, edge.arrow.size=.4, edge.curved=.1)
```



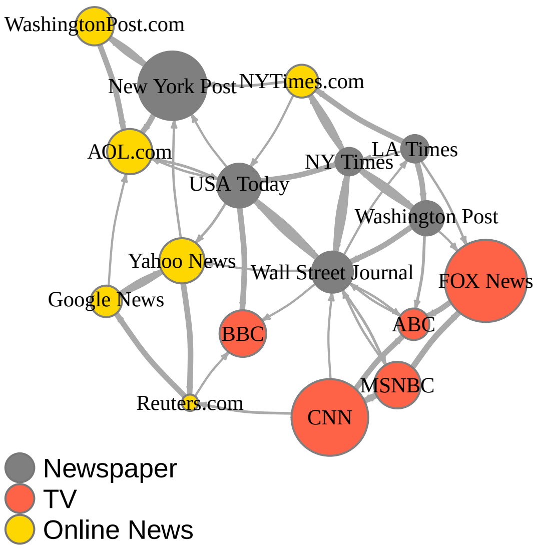
### 3rd plot:

```
# Set edge color to gray, and the node color to orange.
### Replace the vertex label with the node names stored in "media" ###
plot(net, edge.arrow.size=.2, edge.curved=.1,
     vertex.color="orange", vertex.frame.color="gray50",
     vertex.label=V(net)$media, vertex.label.color="black", vertex.label.cex=.7)
```



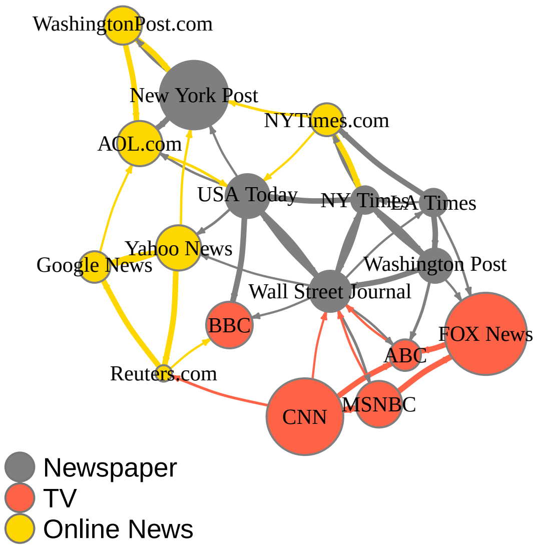
### 4th plot:

```
##### Generate colors based on media type: #####
colrs <- c("gray50", "tomato", "gold")
V(net)$color <- colrs[V(net)$media.type]
##### Set node size based on audience size: #####
V(net)$size <- V(net)$audience.size*0.7
##### Set edge width based on weight: #####
E(net)$width <- 1+E(net)$weight/12
{
##### some attribute you can change #####
# The labels are currently node IDs.
# Setting label to NA will render no labels:
V(net)$label.color <- "black"
V(net)$label <- NA
#change arrow size and edge color:
E(net)$arrow.size <- .2
E(net)$edge.color <- "gray80"
}

plot(net, edge.arrow.size=.2, edge.curved=.1,
     vertex.frame.color="gray50", vertex.label=V(net)$media,
     vertex.label.color="black", vertex.label.cex=.7)
##### add legend #####
legend(x=-1.5, y=-1.1, c("Newspaper","TV","Online News"), pch=21,col="#777777",
       pt.bg=colrs,pt.cex=2, cex=.8,bty="n",ncol=1)
```

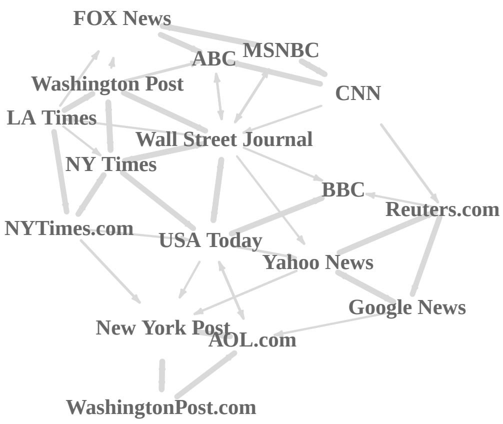


### 5th plot:

```
##### color the edge based on their source node color #####
# get the starting node for each edge with the ends()
edge.start <- ends(net, es=E(net), names=F)[,1]
edge.col <- V(net)$color[edge.start]

plot(net, edge.arrow.size=.2, edge.curved=.1, edge.color = edge.col,
     vertex.frame.color="gray50", vertex.label=V(net)$media,
     vertex.label.color="black", vertex.label.cex=.7)
legend(x=-1.5, y=-1.1, c("Newspaper","TV","Online News"), pch=21,col="#777777",
       pt.bg=colrs,pt.cex=2, cex=.8,bty="n",ncol=1)
```



```
# Sometimes, especially with semantic networks,
# we may be interested in plotting only the labels of the nodes
plot(net, vertex.shape="none", vertex.label=V(net)$media,
     vertex.label.font=2, vertex.label.color="gray40",
     vertex.label.cex=.7, edge.color="gray85")
```



Reference: Network Analysis and Visualization with R and igraph [Katherine Ognyanova, www.kateto.net]